REMARKS

Claims 1-4, 20-22 and 39-41, and 43 are currently pending in the application. Claims 5-19 and 23-38 have been withdrawn, and claim 42 is cancelled. Claims 1-4, 20-22, and 40-41 stand rejected under 35 U.S.C. §102(e) based on U.S. Patent Application No. 2003/0023744 to Sadot et al. (hereinafter "Sadot"). Claim 39 stand rejected under 35 U.S.C. §103(a) based on Sadot in view of U.S. Patent No. 6,351,812 to Datar *et al.* (hereinafter "Datar"). Claims 42-43 stand rejected under 35 U.S.C. §103(a) based on Sadot in view of U.S. Patent Application No. 2002/0199014 to Yang *et al.* (hereinafter "Yang"). Claims 1, 2, 3, 20, 21, 41 and 43 are amended herein.

More specifically, claim 1 has been amended to recite "content-aware" and "non-content-aware" service requests, rather than a "first type" and a "second type" of service requests, and first logic that is shared by and supports both content-aware and non-content-aware service requests, as follows:

> "when a content-aware service request is received, the first logic determines if the at least one first persistence policy is applicable;
>
> when a non-content-aware service request is received, the first logic determines if the at least one second persistence policy is applicable;
>
> when the content-aware service request is received but it is determined that the at least one first persistence policy is inapplicable, the first logic determines if the at least one second persistence policy is applicable;
>
> wherein the first logic is shared by and supports both content-aware and non-content-aware resource requests;"

The support for these amendments is provided, for example, by the following:

page 19, line 30, to page 20, line 1, of the application, describing an embodiment where the system "supports both L4 and L5-7 transactions through the same mechanism;"

page 19, line 7, of the application, referring to an L4 service request as "non content aware," and referring to a L5-7 service request as "content aware;"

page 21, lines 11-13, and page 28, lines 15-17, describing, for an L5-7 request, checking for persistence using a cookie-based or session-based persistence policy, and if that is unsuccessful,

then checking for persistence using a client-based persistence policy, the same persistence policy used to check for persistence of the L4 service request.

As amended, claim 1 provides efficient handling of content-aware and non-content-aware requests because it recites first logic that supports and is shared by both types of requests.

In contrast to this, Sadot fails to teach or suggest logic that supports and is shared by content-aware and non-content-aware requests, and, in fact, provides no teachings at all regarding non-content-aware requests. To see this, consider Fig. 2 of Sadot, reproduced, below, which the Examiner considered applicable to claim 1, prior to the amendments herein:

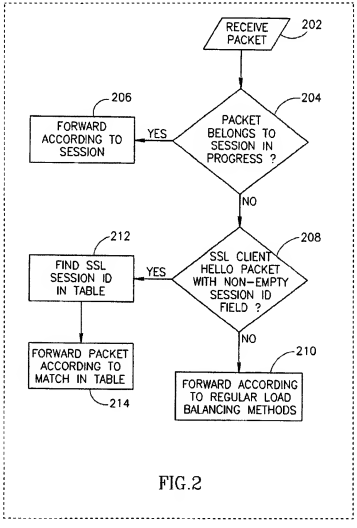


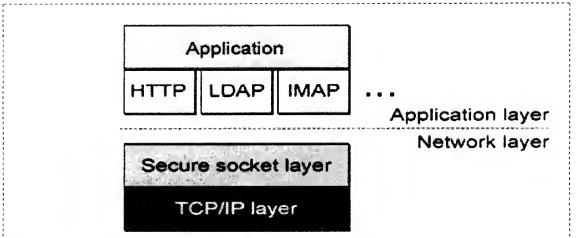
FIG.2

More specifically, prior to the amendments herein, the Examiner considered a packet that belongs to a session in progress, referred to in diamond 204, as the "first type" of request, the policy of forwarding such a packet according to session, referred to in box 206, as the at least one persistence policy, a packet that belongs to a new session, *i.e.*, the SSL client HELLO packet with a non-empty session ID field, referred to in diamond 208, as the "second type" of request, and the policy of forwarding such a packet according to session ID, referred to in box 214, as the at least one second persistence policy. (*See* Office Action, pages 3-4).

Of critical importance is that while the Examiner considered a packet belonging to a new session as corresponding with the second type of request, the Examiner must have implicitly considered such a packet to also correspond to the first type of request where it has been determined that the at least one persistence policy is inapplicable. That is the only way Fig. 2 of Sadot could have been considered to have met the following requirement of then claim 1:

"when the first type of service request is received but it is determined that the at least one first persistence policy is inapplicable, the first logic determines if the at least one second persistence policy is applicable;"

However, with the amendments herein to claim 1, such a strained application of Fig. 2 of Sadot is no longer possible. That is because, with the amendments herein, Fig. 2 of Sadot cannot be applied to claim 1 because neither a packet that belongs to an existing session, referred to in diamond 204, nor a packet that belong to a new session, referred to in diamond 208, is a non-content-aware request. To see this, consider that both types of packets require awareness of the SSL session of the packet, whether existing or new, and, as Sadot teaches, the SSL protocol is at OSI layer 5, in that it mediates between the TCP layer (OSI layer 4) and the application layer (OSI layer 7). (*See* Sadot, paragraph [0004] ("One common encryption protocol is the secure sockets layer (SSL) protocol which mediates between application protocols (e.g., HTTP, FTP) and a transport protocol (e.g., TCP).")). The following figure, from a CryptoHeaven definition of "SSL protocol," attachéd hereto, also shows that the SSL protocol runs at OSI layer 5, *i.e.*, above the TCP protocol (OSI layer 4) and below high-level application protocols (OSI layer 7):



SSL runs above TCP/IP and below high-level application protocols.

Since both packets run at OSI layer 5, neither packet is non-content-aware according to the lexicography of the specification.[1] (*See* page 19, line 7 (referring to "an L4 (non content aware) or L5-7 (content aware) service")). Since neither type of packet shown in Fig. 2 of Sadot is non-content-aware, Sadot does not meet the following element of claim 1:

"when a non-content-aware service request is received, the first logic determines if the at least one second persistence policy is applicable;"

And, as stated at the outset, Sadot certainly does not meet the requirement or provide the benefit of first logic that is "shared by and supports both content-aware and non-content-aware resource requests. . . ."

Based on the foregoing, the Examiner's claim, at page 7 of the Office Action, that Sadot teaches both content-aware and non-content-aware requests, is, Applicant respectfully submits, unfounded.

Datar and Yang do not fill this gap in teaching in Sadot. As a result, claim 1 is patentable over Sadot, Datar and Yang, considered singly and in combination.

All the other independent claims, *i.e.*, claims 3, 20, 21 and 41, have been amended to recite the same or similar requirements as claim 1. Hence, all claims are patentable over Sadot, Datar and Yang, considered singly and combination.

[1] Applicant's lexicography controls over any contrary interpretation. (*See* MPEP §2111.01(IV)).

For all the foregoing reasons, reconsideration of and withdrawal of all outstanding rejections is respectfully requested. The Examiner is earnestly solicited to allow all claims, and pass this application to issuance.

To expedite allowance of this case, the Examiner is earnestly invited to call Robert C. Laurenson at (949) 759-5269.

Respectfully submitted,

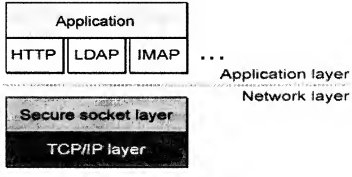
Date: December 19, 2007

/Robert C. Laurenson/
Robert C. Laurenson (Reg. No. 34,206)

HOWREY LLP
2941 Fairview Park Drive, **Box No. 7**
Falls Church, VA 22042
FAX No. (703) 336-6950
Telephone No. (949) 759-5269

SSL Protocol

The Transmission Control Protocol/Internet Protocol (TCP/IP) governs the transport data over the Internet. Other protocols, such as the HyperText Transport Prc Lightweight Directory Access Protocol (LDAP), or Internet Messaging Access Protoc "on top of" TCP/IP in the sense that they all use TCP/IP to support typical application displaying web pages or running email servers.



SSL runs above TCP/IP and below high-level application protocols.

The SSL protocol runs above TCP/IP and below higher-level protocols such as HT uses TCP/IP on behalf of the higher-level protocols, and in the process allows ar server to authenticate itself to an SSL-enabled client, allows the client to authentica server, and allows both machines to establish an encrypted connection.

These capabilities address fundamental concerns about communication over the Inte TCP/IP networks:

- SSL server authentication allows a user to confirm a server's identity. SSL- software can use standard techniques of public-key cryptography to check t certificate and public ID are valid and have been issued by a certificate authorit the client's list of trusted CAs. This confirmation might be important if the user, sending a credit card number over the network and wants to check the rec identity.
- SSL client authentication allows a server to confirm a user's identity. Us techniques as those used for server authentication, SSL-enabled server softw that a client's certificate and public ID are valid and have been issued by a certi (CA) listed in the server's list of trusted CAs. This confirmation might be ir server, for example, is a bank sending confidential financial information to a wants to check the recipient's identity.
- An encrypted SSL connection requires all information sent between a client and encrypted by the sending software and decrypted by the receiving software, th high degree of confidentiality. Confidentiality is important for both parties transaction. In addition, all data sent over an encrypted SSL connection is pr mechanism for detecting tampering--that is, for automatically determining wh has been altered in transit.

The SSL protocol includes two sub-protocols: the SSL record protocol and the S: protocol. The SSL record protocol defines the format used to transmit data. The S

protocol involves using the SSL record protocol to exchange a series of messages be
enabled server and an SSL-enabled client when they first establish an SSL co
exchange of messages is designed to facilitate the following actions:

- Authenticate the server to the client.
- Allow the client and server to select the cryptographic algorithms, or ciphers, support.
- Optionally authenticate the client to the server.
- Use public-key encryption techniques to generate shared secrets.
- Establish an encrypted SSL connection.

SSL technology is used to establish a secure and encrypted communication channe
Internet connected devices.

Next section: Transparent Encryption Technology.